

# RaceTrack2 - RESPONSABILITA'

- **Controller:**  
Controller: Gestisce il gioco, crea la gara e controlla ogni singola azione del gioco
- **View:**  
PromptGUI: Permette di muovere il gioco aspettando gli input dell'user
- **Model:**
  - **points:**  
Point: si occupa di gestire le coordinate x e y entrambe positive presenti nel piano.  
TrackPoint: classe che estende Point, si occupa di gestire oltre alla locazione del punto anche il tipo del punto.
  - **track:**  
Track: si occupa di gestire il circuito di gara.
  - **vector:**  
Vector: utilizzata per permettere alla macchina di muoversi, si occupa di gestire la lunghezza di un vettore, dotati di un inizio e una fine.
- **car:**
  - Car: si occupa di gestire tutte quelle caratteristiche indispensabili alla macchina, tra cui movimento, colore, velocità e il primo vettore.
- **bot:**
  - Bot: si occupa di creare e gestire il movimento e lo stato di un giocatore, in particolare il nome, la macchina, lo stato del giocatore e il round.
- **engine:**
  - SystemMovement: classe utilizzata per creare il nuovo vettore di movimento di un giocatore in relazione al suo ultimo vettore.
  - GameInformation: classe utilizzata per stampare le informazioni utili di ogni giocatore durante la gara
  - RuleChecker: gestisce le informazioni da stampare riferite al movimento e allo stato di ogni bot nel caso in cui: un bot abbiamo finito il circuito, sia uscito dal circuito o se è andato a sbattere.
- **loader:**
  - LoaderFile: classe utilizzata per creare e scrivere (salvare) il gioco da un file json
- **Game:** classe che permette di gestire la gara, la sua conclusione, il suo corretto svolgimento e la relazione tra giocatori e circuito
- **main:** App: main